Multi-Level Contextual RNNs With Attention Model for Scene Labeling

Heng Fan, Xue Mei, Danil Prokhorov, and Haibin Ling

***Abstract*— Image context in image is crucial for improving scene labeling. While the existing methods only exploit local context generated from a small surrounding area of an image patch or a pixel, the long-range and global contextual information is often ignored. To handle this issue, we propose a novel approach for scene labeling by multi-level contextual recurrent neural networks (RNNs). We encode three kinds of contextual cues, viz., local context, global context, and image topic context in structural RNNs to model long-range local and global dependencies in an image. In this way, our method is able to "see" the image in terms of both long-range local and holistic views, and make a more reliable inference for image labeling. Besides, we integrate the proposed contextual RNNs into hierarchical convolutional neural networks, and exploit dependence relationships at multiple levels to provide rich spatial and semantic information. Moreover, we adopt an attention model to effectively merge multiple levels and show that it outperforms average- or max-pooling fusion strategies. Extensive experiments demonstrate that the proposed approach achieves improved results on the CamVid, KITTI, SiftFlow, Stanford Background, and Cityscapes data sets.**

***Index Terms*— Scene labeling, scene understanding, contextual recurrent neural networks (CRNNs), attention model, intelligent transportation system.**

I. Introduction

SCENE labeling, also known as semantic segmentation, refers to assigning one of semantic classes to each pixel in an image, which plays an important role in intelligent vehicles since vehicles need to analyze and understand environments around them. For example, the vehicles must be able to discriminate building, car, pedestrian, road and so on in a traffic scene (see Figure 1). To address this problem, a large body of researches [1], [5], [8], [17], [27], [31], [40], [43]–[45], [50], [51], [57] have been done on scene labeling.

For image labeling, knowledge of a long-range context is crucial. However, existing methods mainly focus on exploiting short-range context. They are prone to misclassify visually similar pixels which actually belong to different classes. For example, 'sand' and 'road' pixels are hard to distinguish with limited short-range context. However, if we consider long-range context for 'sand' ('water' pixels) and 'road' ('grass' pixels) pixels, their differentiations become conspicuous.

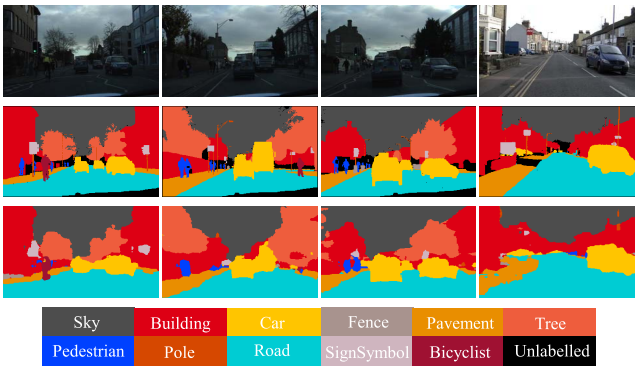

Fig. 1. Some quantitative labeling results on CamVid. **First row:** input images. **Second row:** groundtruth. **Third row:** our prediction labels.

Recurrent neural networks (RNNs) [16] have been successfully applied to natural language processing (NLP) [18], [20] owing to the capability of encoding long-range contextual information in sequential data. There are attempts to bring RNNs to computer vision community [4], [8], [14], [15], [19], [37], [45], [53], [62]. Among them, [45] proposes the graphical RNNs to model long-range dependencies among image units.

Inspired by this idea, we present the multi-level contextual RNNs for scene labeling. Specifically, we incorporate three kinds of contextual cues, i.e., local context, global context and image topic context in structural RNNs to model long-range local and global dependencies among image units. For local context, we consider eight neighbors for each image unit. Different from previous methods, this local context is encoded in RNNs, and the local contexts of all image units are thus connected in a structural undirected cyclic graph, which leads to long-range local contexts in images. However, conventional RNNs are utilized to handle temporal data and not suitable to be directly applied to spatially distributed data. We thus decompose the structural undirected cyclic graph into directed acyclic graphs as in [45]. Differently from [45], we consider assigning different weights to the neighbors of each image unit because different neighbors play different roles in inference. For example, the neighbors whose labels are the same with image unit should play a more important role while others should be assigned less

Manuscript received February 27, 2017; revised July 28, 2017 and October 20, 2017; accepted November 5, 2017. Date of publication January 23, 2018; date of current version November 9, 2018. The Associate Editor for this paper was J. M. Alvarez. *(Corresponding author: Haibin Ling.)*

H. Fan and H. Ling are with the Department of Computer and Information Science, Temple University, Philadelphia, PA 19121 USA (e-mail: hengfan@temple.edu; hbling@temple.edu).

X. Mei and D. Prokhorov are with the Toyota Research Institute, North America, Ann Arbor, MI 48105 USA (e-mail: xue.mei@toyota.com; danil.prokhorov@toyota.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2775628

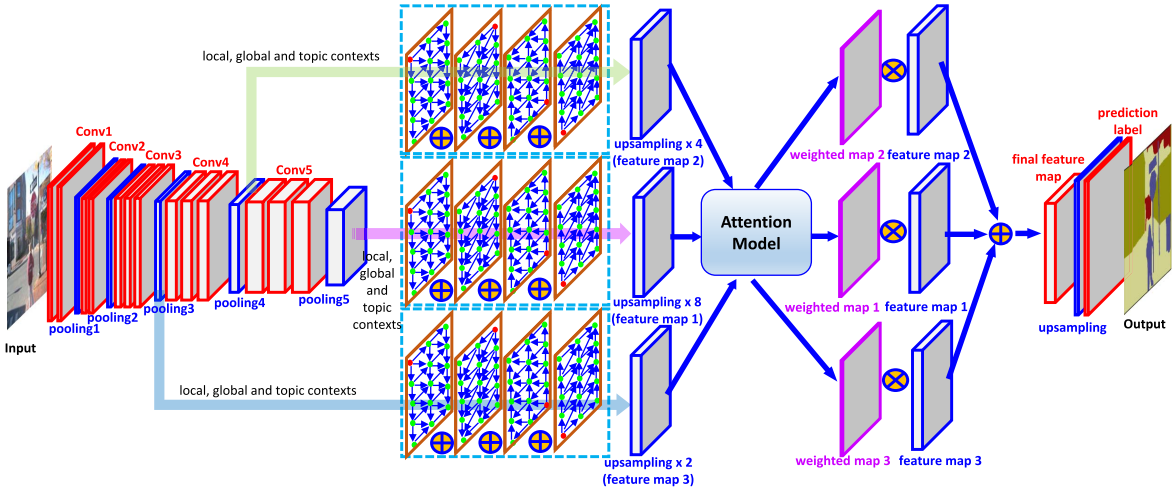

Fig. 2. Illustration of our approach. We adopt CNNs to extract deep features from multiple levels, i.e., the 3^{th}, 4^{th} and 5^{th} pooling layers, which encode rich spatial and semantic information. Four multiple contextual RNNs are utilized to model dependencies at multiple levels. In addition, we use a novel attention model, which takes into account the importance of different levels for pixel classification, to effectively merge these feature maps. With the help of upsampling layers, an end-to-end network is built for image labeling. Note that the image topic features are extracted from input (not shown here for simplicity).

importance. Moreover, we incorporate global and image topic contexts into RNNs which let RNNs 'see' the image in a wider view. Taking advantages of hierarchical convolutional neural networks (CNNs) into account, we integrate our contextual RNNs into CNNs, and utilize dependencies in multiple levels to provide rich spatial and semantic information. An attention model is adopted to effectively fuse these multiple levels, and we show the benefits of attention model over two common fusion strategies. Integrating CNNs with RNNs, we propose an end-to-end network as shown in Figure 2. Experiments on five challenging benchmarks showcase the effectiveness of our approach.

In summary, we make the following contributions:

- We propose the contextual RNNs which encode three kinds of contextual cues to model long-range dependencies in an image for scene labeling.
- We use different dependencies in multiple levels by integrating RNNs and CNNs to provide rich spatial and semantic information for image labeling. Besides, a novel attention model is adopted to improve effectiveness.
- Experiments on CamVid [3], KITTI [21], SiftFlow [27], Stanford-background [24] and Cityscapes [34] show that our method outperforms other scene labeling approaches.

II. Related Work

As a fundamental task in computer vision, image labeling has attracted increasing attention in recent years. Early non-parametric approaches try to transfer labels of training data to query images and perform label inference in a probabilistic graphical model (PGM). Liu *et al.* [27] propose a non-parametric image parsing method by estimating 'SIFT Flow' between images, and infer the labels of pixels in a markov random field (MRF). Krähenbühl and Koltun [26] build a fully connected graph to incorporate higher order dependencies among image units. Tighe and Lazebnik [50] introduce a superparsing method for scene parsing. Yang *et al.* [57] suggest to incorporate context information to improve image retrieval and superpixel classification for semantic labeling. Reference [24] presents a approximate nearest neighbor algorithm for semantic segmentation by developing a structured graph over superpixels. Álvarez *et al.* [1] propose to model an image set as a fully connected pairwise conditional random field (CRF) [26] defined over image units (pixels or superpixels) with Gaussian edge potentials for efficient scene parsing. Reference [51] proposes to combine per-exemplar sliding window detector for image parsing task. Despite promising results for scene labeling, the above methods only use hand-crafted features for image representation, which is less robust in complex scene.

The CNNs [32], which demonstrate the power in extracting high-level feature representations [46], have been used for scene labeling. Farabet *et al.* [17] propose to learn hierarchical features with CNNs for scene labeling. Long *et al.* [29] introduce the fully conventional networks (FCN) for semantic labeling. In addition, to incorporate more spatial information, they adopt a skip strategy to fuse features from multiple levels. Ghiasi and Fowlkes [22] propose a multi-resolution reconstruction approach based on a Laplacian pyramid to refine segment boundaries in scene parsing. Shuai *et al.* [44] adopt CNNs as parametric model to learn discriminative features and integrate it with a non-parametric model to infer pixel labels. Pinheiro and Collobert [40] utilize CNNs in a recurrent way to model spatial dependencies in image by attaching raw input with output of CNNs. Wang *et al.* [55] propose a joint approach of priori convolutional neural networks at superpixel level and soft restricted context transfer for street scene labeling. Gao *et al.* [23] suggest to embed contour information and location prior for segmentation. Liang *et al.* [31] suggest to model relationships among intermediate layers with RNNs for scene labeling. Badrinarayanan *et al.* [6] propose an encoder-decoder architecture for semantic segmentation. a similar framework

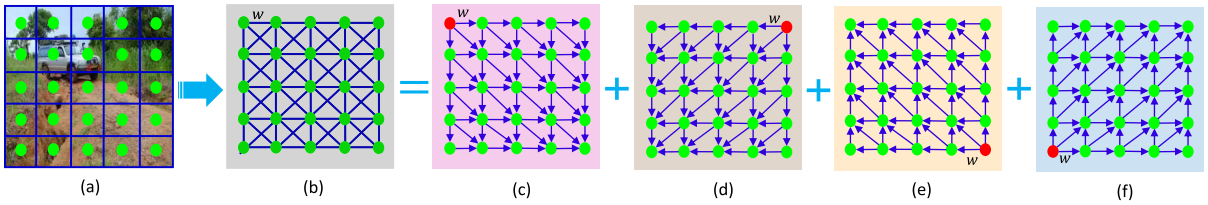

Fig. 3. Decomposition of undirected cyclic graph into four directed acyclic graphs. The green solid dots represent image units, red solid dots are start points in directed acyclic graphs, and w denotes the weight. Note that the inputs for RNNs are pooling layers from CNNs, and here we just illustrate this process. (a) input. (b) undirected cyclic graph. (c) directed acyclic graph (southeast). (d) directed acyclic graph (southwest). (e) directed acyclic graph (northwest). (f) directed acyclic graph (northeast).

is presented in [35]. However, they do not consider long-range dependencies among image units for image labeling. To alleviate this issue, Zheng *et al.* [60] propose to utilize CRFs as recurrent neural networks to improve the capacity of CNNs in delineating objects. Different from [60], Lin *et al.* [28] propose to use CRFs to learn the patch-patch context in CNNs for scene labeling. Yu and Koltun [56] propose to improve contextual information in CNNs by expanding receptive fields of filters and present the dilated convolutional networks for dense segmentation. To the same end, Chen *et al.* [11] present atrous convolution and combine it with the fully connected CRFs for semantic segmentation.

Recently, RNNs have drawn increasing attention in computer vision owing to the capability of capturing long-range contextual information. Oord *et al.* [37] propose to model discrete probability of raw pixel values with RNNs for image completion. Graves *et al.* [19] extend one dimensional RNNs to multi-dimensional RNNs for handwriting recognition. Based on [19], Byeon *et al.* [8] propose a two-dimensional long-short term memory (LSTM) for scene parsing. This method is able to model long-range local context in image. Visin *et al.* [53] propose to use RNNs to model structured information of local generic features extracted from CNNs for segmentation.

The most relevant works to ours are [62] and [45]. Zuo *et al.* [62] propose to use hierarchical two dimensional RNNs to model spatial dependencies among image regions from multiple scales, and concatenate these dependencies for image classification. Shuai *et al.* [45] use graphical RNNs to model long-range context in image for scene labeling. Though [45] and [62] both utilize RNNs, our work is different from them in three aspects: (1) Considering different levels of importance of different neighbors for each image unit, we assign different weights to each of the neighbors. In doing so, we propose the weighted structural RNNs. However, both [45] and [62] treat all neighbors of an image unit the same. (2) For image labeling, global and topic contexts also play crucial roles in distinguishing pixels. We propose the contextual RNNs by incorporating local, global and topic contexts into structural RNNs. Our contextual RNNs are able to capture both long-range local and global dependencies among image units and thus see the entire image in a wider view. Nevertheless, [45] and [62] only take the local context into consideration in RNNs. (3) To exploit rich spatial and semantic information, we integrate the contextual RNNs with CNNs and utilize various dependencies in multiple levels. An attention model is adopted to merge information from these multiple levels. However, [45] only models the dependencies among image units in one layer (the 5^th pooling layer). In [62], RNNs are used to model spatial dependencies among image regions from one layer with multiple scales, and the outputs of different scales are simply concatenated, which is different from ours using the attention model to combine features from different layers.

III. The Proposed Approach

A. Basic Recurrent Neural Networks (RNNs)

RNNs are developed to model dependencies in temporally ordered data. Specifically, the hidden layer $h^{(s)}$ in RNNs at time step s is represented by a non-linear function over current input $x^{(s)}$ and hidden layer at previous time step $h^{(s-1)}$. The output layer $y^{(s)}$ is connected to hidden layer $h^{(s)}$.

Given an input sequence $\{x^{(s)}\}_{s=1,2,\cdots,S}$, the hidden and output layers at each time step s are computed with

$$\begin{cases} h^{(s)} = \phi(Ux^{(s)} + Wh^{(s-1)} + b_h) \\ y^{(s)} = \sigma(Vh^{(s)} + b_y) \end{cases} \tag{1}$$

where U, W and V denote shared weight matrices between input and hidden layer, previous hidden layer and current hidden layer, and hidden layer and output layer respectively. b_h and b_y are two bias terms, and $\phi(\cdot)$ and $\sigma(\cdot)$ are non-linear activation functions. Since the inputs are progressively stored in hidden layers, RNNs thus can keep such "distributed" memory and model long-range dependencies in sequence.

B. Contextual Recurrent Neural Networks (CRNNs)

Different from RNNs, our CRNNs encode three contextual cues which are local, global and topic contexts. This section will introduce the incorporation of these contexts, and forward and backward operations of CRNNs.

1) Local Context: One of our goals is to model long-range context in image. For an image, the interactions among image units can be represented as an undirected cyclic graph (see Figure 3(b)). Due to loopy structure of undirected cyclic graph, however, the aforementioned basic RNNs cannot be directly applied to images. To address this issue, we approximate the topology of undirected cyclic graph by the combination of several directed acyclic graphs as in [45], and use variant RNNs to model long-range local context in these

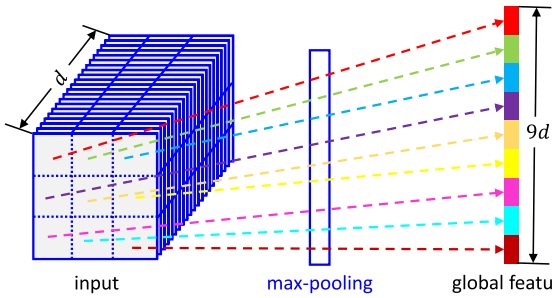

Fig. 4. Illustration of extracting global feature. The d is number of channel of input. The extracted global feature can capture more context information.

directed acyclic graphs as shown in Figure 3. For each directed acyclic graph, the main difference is the position of start point.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ denote the directed acyclic graph, where $\mathcal{V} = \{v_i\}_{i=1,2,\cdots,N}$ denotes vertex set and $\mathcal{E} = \{e_{ij}\}$ is edge set in which e_{ij} represents directed edge from v_i to v_j. The structure of RNNs follows the same topology as $\mathcal{G}$. A forward propagation sequence can be seen as traversing $\mathcal{G}$ from start point, and each vertex relies on its all predecessors. For vertex v_i, therefore, the hidden layer $h^{(v_i)}$ is expressed as a non-linear function over current input $x^{(v_i)}$ at v_i and summation of hidden layers of all its predecessors. Specifically, the hidden layer $h^{(v_i)}$ and output layer $y^{(v_i)}$ at each v_i are computed with

$$\begin{cases} h^{(v_i)} = \phi\ (Ux^{(v_i)} + W \sum_{v_j \in \mathcal{P}_\mathcal{G}(v_i)} h^{(v_j)} + b_h) \\ y^{(v_i)} = \sigma(Vh^{(v_i)} + b_y) \end{cases} \tag{2}$$

where $\mathcal{P}_\mathcal{G}(v_i)$ denotes the predecessor set of v_i in $\mathcal{G}$. In [45], the recurrent weight matrix W is shared across all the predecessors of v_i. For v_i, nevertheless, different predecessors should be assigned with different weights. For example, the predecessors whose labels are the same with v_i may be more important in inferring the label of v_i while others play less important roles. Thus we revise Eq (2) as follows

$$\begin{cases} h^{(v_i)} = \phi\ (Ux^{(v_i)} + \sum_{v_j \in \mathcal{P}_\mathcal{G}(v_i)} W^{(v_j)} h^{(v_j)} + b_h) \\ y^{(v_i)} = \sigma(Vh^{(v_i)} + b_y) \end{cases} \tag{3}$$

where $W^{(v_j)}$ denotes the weight matrix of predecessor v_j. With Eq (3), RNNs can model long-range context in images.

2) Global Context: To further improve the ability of RNNs for pixel classification, we also consider global context in RNNs. For the input (i.e., pooling layer in CNNs), we first partition it into 3×3 blocks. Then max-pooling is performed on each block. Such partition and max-pooling result in nine feature vectors, which are concatenated as a global feature for input. Figure 4 illustrates the extraction of the global feature.

Let $g = [g_1, g_2, \cdots, g_9]^T$ denote the global feature, where g_i represents feature obtained by max-pooling over block i. To incorporate global contextual information into RNNs, we revise Eq (3) as the following

$$\begin{cases} h^{(v_i)} = \phi\ (Ux^{(v_i)} + \sum\limits_{v_j \in \mathcal{P}_\mathcal{G}(v_i)} W^{(v_j)} h^{(v_j)} + Gg + b_h) \\ y^{(v_i)} = \sigma(Vh^{(v_i)} + b_y) \end{cases} \tag{4}$$

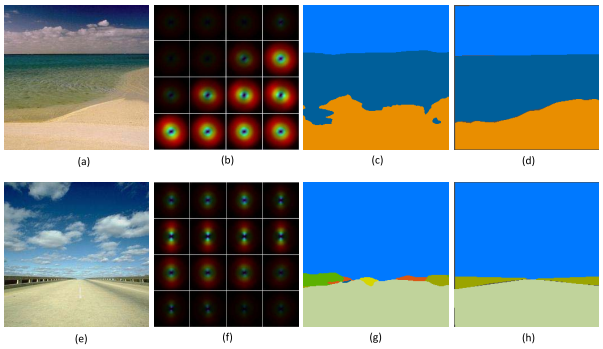

Fig. 5. Visualization of GIST feature. Images (a) and (e) are inputs, (b) and (f) are their topic features. With these topic contexts, our RNNs are able to distinguish similar pixels. Images (c) and (g) are our results, and (d) and (h) are the groundtruth.

where G is the recurrent weight matrix for global feature g. Through Eq (4), the RNNs can capture both long-range local and global contextual information in the image.

3) Image Topic Context: Long-range local and global contexts can help to distinguish visually similar pixels. However, for some situations, it is still hard to classify pixels only with these two kinds of contexts. To further improve the ability of RNNs to distinguish pixels, we propose to incorporate topic context information into RNNs. For instance, the 'sand' pixels in Figure 5(a) and 'road' pixels in Figure 5(e) are very similar. The dependencies of local features are limited to correctly classify these two kinds of pixels. Besides, most part of these two images are semblable which results in confused global context features of these two images. The context obtained by combining local and global features is too strong to distinguish similar pixels and may result in worse consequence. However, the topic features of these two images are different (see Figure 5(b) and Figure 5(f)). If the RNNs know their topic features, it will be easier to discriminate these similar pixels.

In this paper, we adopt GIST feature [38] as the topic feature. GIST feature represents holistic abstraction of an image, and has been applied to recognition [12], [54] and image retrieval [61]. For our network, the GIST feature is extracted from raw input image, denoted as t. To encode topic context, we revise Eq (4) as follows

$$\begin{cases} h^{(v_i)} = \phi\ (Ux^{(v_i)} + \sum\limits_{v_j \in \mathcal{P}_\mathcal{G}(v_i)} W^{(v_j)} h^{(v_j)} + Gg + Tt + b_h) \\ y^{(v_i)} = \sigma(Vh^{(v_i)} + b_y) \end{cases} \tag{5}$$

where t is topic feature extracted from raw input and T denotes its recurrent weight matrix. Note that topic context is different from global context. The global context encoded in global feature is still pixel-level while topic context encoded in GIST feature is image-level.

By incorporating local, global and topic contexts, our CRNNs are able to model the dependencies among image units in a wider view and thus become better at classifying pixels.

4) Forward and Backward of CRNNs: The CRNNs are trained via forward pass and backward propagation. With Eq (5), we can compute the forward operation of our CRNNs.

For backward propagation, we need to calculate derivatives at each vertex in the CRNNs. For each vertex in the directed acyclic graph, it is processed in the reverse order of forward propagation sequence. To compute the derivatives at vertex v_i, we need to look at the forward passes of all its successors. Let $\mathcal{S}_{\mathcal{G}}(v_i)$ denote the direct successor set for v_i in $\mathcal{G}$. For each $v_k \in \mathcal{S}_{\mathcal{G}}(v_i)$, its hidden layer is computed with

$$\begin{cases} h^{(v_k)} = \phi\ (Ux^{(v_k)} + W^{(v_i)}h^{(v_i)} + \mathcal{M} + Gg + Tt + b_h) \\ y^{(v_k)} = \sigma(Vh^{(v_k)} + b_y) \end{cases} \tag{6}$$

where

$$\mathcal{M} = \sum_{v_l \in \mathcal{P}_{\mathcal{G}}(v_k) - \{v_i\}} W^{(v_l)} h^{(v_l)}$$

Combining Eq (5) and (6), we can see that the errors back-propagated to the hidden layer at v_i come from two sources: directed errors from v_i (i.e., $\frac{\partial y^{(v_i)}}{\partial h^{(v_i)}}$) and a summation over indirected errors from all its successors $v_k \in \mathcal{S}_{\mathcal{G}}(v_i)$ (i.e., $\sum_{v_k} \frac{\partial y^{(v_k)}}{\partial h^{(v_i)}} = \sum_{v_k} \frac{\partial y^{(v_k)}}{\partial h^{(v_k)}} \frac{\partial h^{(v_k)}}{\partial h^{(v_i)}}$). Therefore, the derivatives at vertex v_i can be obtained with

$$\begin{cases} \mathrm{d}h^{(v_i)} = V^T \sigma\prime(y^{(v_i)}) + \sum_{v_k \in \mathcal{S}_{\mathcal{G}}(v_i)} (W^{(v_i)})^T \mathrm{d}h^{(v_k)} \circ \phi\prime(h^{(v_k)}) \\ \nabla W^{(v_i)} = \sum_{v_k \in \mathcal{S}_{\mathcal{G}}(v_i)} \mathrm{d}h^{(v_k)} \circ \phi\prime(h^{(v_k)})(h^{(v_i)})^T \\ \nabla U^{(v_i)} = \mathrm{d}h^{(v_i)} \circ \phi\prime(h^{(v_i)})(x^{(v_i)})^T \\ \nabla G^{(v_i)} = \mathrm{d}h^{(v_i)} \circ \phi\prime(h^{(v_i)})(g)^T \\ \nabla T^{(v_i)} = \mathrm{d}h^{(v_i)} \circ \phi\prime(h^{(v_i)})(t)^T \\ \nabla b_h^{(v_i)} = \mathrm{d}h^{(v_i)} \circ \phi\prime(h^{(v_i)}) \\ \nabla V^{(v_i)} = \sigma\prime(y^{(v_i)})(h^{(v_i)})^T \\ \nabla b_y^{(v_i)} = \sigma\prime(y^{(v_i)}) \end{cases} \tag{7}$$

where $\circ$ represents the Hadamard product, $\sigma\prime(\cdot) = \frac{\partial L}{\partial y(\cdot)} \frac{\partial y(\cdot)}{\partial \sigma}$ is the derivative of loss function L with respect to output function σ, and $\phi\prime(\cdot) = \frac{\partial h}{\partial \phi}$. We utilize the average cross entropy loss function to compute L. Note that the superscript T denotes transposition operation.

With Eq (5) and (7), we can perform forward and backward passes on one directed acyclic graph. In this paper, we decompose the undirected cyclic graph into four directed acyclic graphs along southeast, southwest, northwest and northeast directions as in [45]. Figure 3 visualizes the decomposition. Let $\mathcal{G}^{\mathcal{U}} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4\}$ denote the undirected cyclic graph, where $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$ represent the four directed acyclic graphs respectively. For each $\mathcal{G}_m$ ($m = 1, 2, \cdots, 4$), we can get the corresponding hidden layer h_m by performing our CRNNs. The summation of all hidden layers are fed to output layer. We use Eq (8) to express this process

$$\begin{cases} h_m^{(v_i)} = \phi\ (U_m x^{(v_i)} + \sum_{v_j \in \mathcal{P}_{\mathcal{G}_m}(v_i)} W_m^{(v_j)} h_m^{(v_j)} \\ \qquad\qquad + Gg + Tt + b_{h_m}) \\ y^{(v_i)} = \sigma\ (\sum_{\mathcal{G}_m \in \mathcal{G}^{\mathcal{U}}} V_m h_m^{(v_i)} + b_y) \end{cases} \tag{8}$$

where U_m, $W_m^{(v_j)}$, G, T, V_m, and b_{h_m} are matrix parameters and bias term for $\mathcal{G}_m$, b_y is the bias term for final output, and $\mathcal{P}_{\mathcal{G}_m}(v_i)$ denotes the predecessor set of v_i in $\mathcal{G}_m$. Note that the global and topic contexts are shared across four directed acyclic graphs. With Eq (8), we can compute loss L as follows

$$L = -\frac{1}{N} \sum_{v_i \in \mathcal{G}^{\mathcal{U}}} \sum_{j=1}^{C} \log(y_j^{(v_i)} Y_j^{(v_i)}) \tag{9}$$

where N denotes the number of image units, C is the number of classes, $y^{(v_i)}$ represents class likelihood vector and $Y^{(v_i)}$ is the binary label indicator vector for image unit at v_i. The error generated at v_i is computed with

$$\nabla x^{(v_i)} = \sum\nolimits_{\mathcal{G}_m \in \mathcal{G}^{\mathcal{U}}} U_m^T \mathrm{d}h_m^{(v_i)} \circ \phi\prime(h_m^{(v_i)}) \tag{10}$$

C. Multi-Level Contextual RNNs With Attention Model

We integrate our CRNNs into CNNs to model dependencies in intermediate layers. In CNNs, different layers possess various information. The high-level layers capture more semantic information, whereas low-level layers encode more spatial information. For scene labeling, both semantic and spatial pieces of information are crucial. Therefore, we use CRNNs to exploit dependencies in multiple levels and combine them to provide rich semantic and spatial information for scene labeling. To fuse these multiple levels, average-pooling [9], [13] and max-pooling [39] are two simple and common strategies. However, different levels with different scales contain various contexts. In this paper, we propose to adopt the attention model [2], [10] to exploit the importance of different levels. In [2], attention model is used to softly weight the importance of words in a source sentence when predicting a target word, and [10] adopts attention model to weight different input data. In our work, we use attention model to weight multiple levels.

Specifically, let $\{f_{i,c}^q\}_{q=1,2,\cdots,Q}$ denote Q feature maps of Q levels, where i ranges over all the spatial positions and $c \in \{1, 2, \cdots, C\}$. Note that in our work, the feature maps from pooling layers go through CRNNs and thus have the same number of classes. All the feature maps are resized to have the same resolution via upsampling operation [29]. We denote $z_{i,c}$ to be weighted sum of feature maps at (i, c) for all levels as follows

$$z_{i,c} = \sum\nolimits_{q=1}^{Q} \omega_i^q \cdot f_{i,c}^q \tag{11}$$

where the weight ω_i^q is calculated with

$$\omega_i^q = \frac{\exp(r_i^q)}{\sum_{e=1}^{Q} \exp(r_i^e)} \tag{12}$$

where r_i^q represents the feature map generated by the attention model at position i for level q. The adopted attention model consists of two convolutional layers. The first layer contains 512 filters with kernel size 3×3 and the second layer has Q filters with kernel size 1×1, where Q denotes the number of levels. The weight ω_i^q demonstrates the importance of feature at position i in level q. As a consequence, the attention model is able to determine how much attention to pay to for features at different positions and levels. Note that the average-pooling and max-pooling are two special cases of our attention model. Specifically, Eq. (11) will become average-pooling if

the weights ω_i^q are replaced by $1/Q$, and it will be max-pooling if the summation operation becomes the max operation and $\omega_i^q \equiv 1$ for any i and q. Besides, the attention model can be jointly trained with the networks because it allows a gradient of loss function to be back-propagated [2], [10].

D. Global View of Network Architecture

In this section, we describe the global view of our network architecture. As shown in Figure 2, we adopt multiple CRNNs for features from different levels, and each CRNNs take three kinds of cues as input, which are topic, local and global features. The topic (or GIST) feature is extracted from raw RGB image before feeding it to network, and local and global features are extracted from the corresponding pooling layer. After obtaining these three kinds of features, we feed them to CRNNs and obtain output feature (see Eq. (5)) with same resolution of input pooling layer. With multiple layers, we can obtain various output features. These output features (after upsampling) fused via an attention model to get final output.

IV. Experimental Results

We test our approach on four benchmarks, including three traffic scene datasets (CamVid [3], KITTI [21] and Cityscapes [34]) and two outdoor scene datasets (SiftFlow [27] and Stanford-background [24]). Three metrics, pixel accuracy, class accuracy (i.e., the average of pixel accuracies per class) and mean IoU (Intersection over Union) are adopted to evaluate the performance of our method.

A. Implementation Details

We borrow the architecture and parameters from the VGG-16 network [46] before the 5^{th} pooling layer. Three independent CRNNs are utilized to model image unit dependencies in multiple levels, i.e., the 3^{rd}, 4^{th} and 5^{th} pooling layers. The dimensions of hidden layers of CRNNs are set to the same as the channels of the 3^{rd}, 4^{th} and 5^{th} pooling layers. Non-linear activation function $\phi = max(0, x)$ and σ is *softmax* function. In practice, we apply σ after final upsampling layer (see Figure 2) and use Eq (9) to compute the loss between prediction and groundtruth. The full networks (including CNNs and CRNNs parts) are trained by stochastic gradient descent (SGD) with momentum. The learning rate is initialized to be 10^{-3} and decays exponentially with the rate of 0.9 after 10 epochs. The batch sizes for both training and testing phases are set to 1. The results are reported after 60 training epochs. The entire network is implemented in MATLAB using MatConvNet [52] on a single NVIDIA GTX TITAN Z GPU with 6GB memory.

B. CamVid Dataset

CamVid is a road scene dataset which contains 701 images of day and dusk scenes [3]. Each image is labelled with 11 classes. We follow the usual split protocol [51] (468/233) to obtain training and testing images. Table I demonstrates out results and comparisons with state-of-the-art methods, and Table II shows the individual class accuracy performance.

TABLE I
Quantitative Results (%) and Comparisons on CamVid

Method	Pix. Acc.	Cla. Acc.	Mean IoU
[5]	82.1	62.5	n/a
[30]	83.8	56.1	n/a
[45]	**91.6**	**78.1**%	n/a
[49]	83.8	59.2	n/a
[50]	78.6	43.8	n/a
[51]	83.9	62.5	n/a
[58]	82.1	55.4	n/a
[6]	90.4	71.2	60.1
[53]	88.7	68.1	58.8
[1]	81.6	51.0	n/a
Ours$_{avg}$	90.7	76.6	63.9
Ours$_{max}$	89.9	74.9	64.1
Ours$_{att}$	**91.9**	**77.2**	**66.8**

TABLE II
Individual Class Accuracy (%) Performance on CamVid

Class	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicycle
[49]	84.5	72.6	**97.5**	72.7	34.1	95.3	34.2	45.7	8.1	77.6	28.5
[51]	83.1	73.5	94.6	78.1	48.0	96.0	58.6	32.8	5.3	71.2	**45.9**
[6]	89.6	83.4	96.1	87.7	52.7	96.4	62.2	53.45	32.1	93.3	36.5
[53]	86.8	84.7	93.0	87.3	48.6	**98.0**	63.3	20.9	35.6	87.3	43.5
[1]	76.1	84.3	97.2	57.8	11.0	**98.0**	19.6	0.1	3.3	67.9	45.9
Ours$_{avg}$	90.6	**90.3**	93.9	**95.2**	**68.9**	92.9	67.4	58.5	52.7	94.7	37.3
Ours$_{max}$	88.7	89.5	91.6	93.2	67.5	90.7	66.9	54.9	51.6	93.8	35.9
Ours$_{att}$	**91.3**	89.8	95.2	94.3	67.8	94.0	**68.9**	**59.2**	**53.2**	**96.8**	38.5

As shown in Table I, our method outperforms other approaches on pixel accuracy. However, [45] performs better than our method on class accuracy. We analyse two reasons for this. First, [45] utilizes additional information in the dataset. In [45], the frequency of each class is calculated. Based on the frequency, a weighting function that attends to rare classes is adopted. In this way, the accuracy for non-frequent classes are phenomenally boosted. In this work, we do not use the class balancing strategy but a boost could be expected if we have used it. Second, there is only one scene involved in CamVid (i.e., the road scene). In this situation, the function of global and topic contexts is unconspicuous. Besides, we show that the attention model (Ours$_{att}$) performs better than average-pooling (Ours$_{avg}$) or max-pooling (Ours$_{max}$) strategies.

C. KITTI Dataset

The KITTI dataset [21] contains 252 images of urban scene annotated by [59]. Each image is labelled with 10 semantic classes. We follow the split protocol as in [59] in which 140 images are used for training while the rest for testing. Table III shows our results and comparisons with other approaches.

As shown in Table III, the proposed method outperforms other methods on both pixel and class accuracies. Our method significantly improves the pixel accuracy from 89.3% to 92.1%, and the class accuracy from 65.4% to 70.9%. Though [59] proposes to fuse image and point cloud information and adopts post-processing, our method obtains better results by incorporating context information.

TABLE III
QUANTITATIVE RESULTS (%) PERFORMANCE ON KITTI

Class	Building	Sky	Road	Vegetation	Side	Car	Pedestrian	Cyclist	Sign	Fence	Pix. Acc.	Cla. Acc.	Mean IoU
[59]	95.0	92.6	92.6	92.8	73.3	78.7	65.1	7.3	13.8	**43.2**	89.3	65.4	n/a
[7]	92.5	95.7	92.5	86.3	51.5	67.9	28.6	4.0	2.5	2.3	84.1	52.4	n/a
[29]	90.5	92.5	81.5	97.1	65.5	**85.9**	63.6	4.7	**22.1**	20.0	86.8	62.2	n/a
Ours$_{avg}$	93.4	95.4	85.4	91.1	80.3	84.7	70.2	**35.1**	15.2	36.4	89.6	68.8	57.4
Ours$_{max}$	92.2	96.1	88.9	89.9	79.3	82.2	71.4	34.7	13.7	35.2	88.9	68.4	57.9
Ours$_{att}$	**95.1**	**97.3**	93.3	**95.9**	**82.1**	85.6	**71.5**	34.6	19.6	33.7	**92.1**	**70.9**	**60.1**

TABLE IV
QUANTITATIVE RESULTS (%) AND COMPARISONS ON SIFTFLOW

Method	Pix. Acc.	Cla. Acc.	Mean IoU
[8]	70.11	20.9	n/a
[17]	78.5	29.4	n/a
[27]	74.8	n/a	n/a
[40]	77.7	29.8	n/a
[43]	79.6	33.6	n/a
[44]	80.1	39.7	n/a
[45]	**85.3**	**55.7**	n/a
[57]	79.8	48.7	n/a
[31]	83.5	35.8	n/a
[48]	80.6	45.8	n/a
[36]	83.1	44.3	n/a
[50]	76.9	29.4	n/a
Ours$_{avg}$	85.6	55.9	43.2
Ours$_{max}$	84.3	53.8	42.8
Ours$_{att}$	**86.9**	**57.7**	**44.7**

D. SiftFlow Dataset

The SiftFlow dataset [27] consists of 2688 images captured from 8 typical scenes and annotated with 33 classes. Following the training/testing split protocol in [27], 2488 images are used for training while the rest for testing. The quantitative results and comparisons with state-of-the-art methods are listed in Table IV, and Table V shows the individual class accuracy. Note that only 30 classes appear in the testing images.

As shown in Table IV, our proposed approach outperforms other methods on both pixel and class accuracies. Our ML-CRNNs$_{att}$ can improve the pixel accuracy from 85.3% to 86.9%, and the class accuracy from 55.7% to 57.7%. Though weighting function is adopted to improve performance in [45], our method still achieves better class accuracy because our global and topic contexts are helpful in distinguishing pixels which belong to rare classes.

E. Stanford-Background Dataset

The Stanford-background dataset [24] has 715 images annotated with 8 semantic classes. Following [44] and [48], the dataset is randomly partitioned into 80% (572 images) for training and the rest (143 images) for testing with 5-fold cross validation. As shown in Table VI, the proposed method achieves better results compared with state-of-the-art approaches. Table VII demonstrates the individual class accuracy.

From Table VI, we can see the effectiveness of our ML-CRNNs$_{att}$ with attention model. Both pixel and class accuracies are significantly boosted. The pixel accuracy is improved from 84.6% to 87.2%, and the class accuracy is improved from 77.3% to 78.4%.

F. Cityscapes Dataset

The Cityscapes dataset [34] contains 5000 images of street traffic scene captured from 50 different European cities. Following the training/testing split protocol in [34], 2975 images are used for training, 500 images for validation, and the rest for testing. In total, 19 classes are considered for training and evaluation. The test set ground-truth is withheld by the organizers, and we evaluate the proposed method on their evaluation server.[1] The test results are shown in Table VIII. Figure 6 displays some qualitative labeling results of testing images in the Cityscapes.

Among the compared approaches, [11] and [56] improve context information in networks by expanding the receptive fields of convolution filters. Different from [11] and [56], we adopt the CRNNs to model the dependencies among image units to improve context information. Compared to [11] with mean IoU 70.4% and [56] with mean IoU 67.1%, our approach achieves better performance with mean IoU 71.2%, showing the power of CRNNs.

G. Analysis of Different Context Features

In our ML-RNNs model, we utilize three contexts, i.e., local context, global context and topic context to improve the discriminative ability of RNNs. In order to evaluate the effect of these contexts for the final performance, we develop extra experiments on the four datasets. For each dataset, the baseline experiment only uses the local context for RNNs. Experiments are shown in Table IX.

From Table IX, we can see that compared with the baseline model, both global and topic context features are able to improve the performance of our method, and their roles are different. For datasets Camvid [3] and KITTI [21], all images belong to the same topic (traffic scene). Though incorporation of topic context can improve performance, the global context shows better improvement. For datasets SiftFlow [27] and Stanford-background [24], by contrast, our model with topic context performs better than that with global context because these two datasets contain images with various topics, and topic context captures more discrimination that global context.

[1] https://www.cityscapes-dataset.com/benchmarks/#pixel-level-results.

TABLE V
INDIVIDUAL CLASS ACCURACY (%) PERFORMANCE ON SIFTFLOW

Class	Sky	Building	Tree	Mountain	Road	Sea	Field	Car	Sand	River	Plant	Grass	Window	Sidewalk	Rock
[45]	96.3	90.8	82.1	85.1	89.2	84.8	55.4	84.2	67.9	75.3	51.5	64.8	45.2	63.5	**45.7**
[44]	96.7	88.0	**84.7**	80.3	85.8	75.0	37.6	78.7	37.5	50.1	7.56	**72.6**	33.8	53.7	13.0
[50]	92.0	88.0	84.0	73.0	81.0	85.0	54.0	43.0	28.0	13.0	2.0	49.0	8.0	31.0	4.0
Ours_{avg}	95.2	**91.5**	82.3	83.9	91.1	85.2	59.3	**87.4**	67.0	71.5	45.5	64.8	42.5	66.7	41.2
Ours_{max}	94.1	89.4	81.9	82.1	88.4	82.4	57.9	78.1	65.5	71.2	49.2	60.1	45.3	64.5	42.1
Ours_{att}	**97.8**	90.2	84.2	**85.4**	**91.3**	**86.7**	**62.1**	86.3	**69.3**	**76.3**	**54.2**	63.7	**49.0**	**67.3**	45.3

Class	Bridge	Door	Fence	Person	Staircase	Awning	Sign	Boat	Crosswalk	Pole	Bus	Balcony	Street	Sun	Bird
[45]	37.3	56.8	44.7	36.2	**58.7**	18.3	40.0	63.3	65.2	18.4	1.4	45.8	5.4	**97.9**	0
[44]	**40.2**	44.6	44.2	24.6	14.5	**50.3**	3.05	**74.1**	20.4	7.1	**29.9**	0.05	2.36	86.0	**28.3**
[50]	1.0	6.0	28.0	0	0	0	0	0	15.0	0	0	0	0	1.0	0
Ours_{avg}	33.8	58.4	42.9	**38.7**	53.4	17.3	45.2	64.8	68.1	20.0	2.8	51.2	6.2	95.1	0.04
Ours_{max}	32.3	49.3	41.2	32.3	51.0	17.2	43.5	60.2	66.3	19.3	1.0	50.1	6.1	94.8	0.05
Ours_{att}	36.8	**58.9**	**44.9**	38.1	55.8	19.8	**46.2**	65.7	**69.2**	**23.4**	5.1	**52.1**	**8.2**	89.4	0.04

TABLE VI
QUANTITATIVE RESULTS (%) AND COMPARISONS ON STANFORD-BACKGROUND

Method	Pix. Acc.	Cla. Acc.	Mean IoU
[44]	81.2	71.3	n/a
[48]	**84.6**	**77.3**	n/a
[31]	83.1	74.8	n/a
[8]	78.6	68.8	n/a
[40]	80.2	69.9	n/a
[17]	81.4	76.0	n/a
[25]	79.3	69.4	n/a
[42]	82.9	74.5	n/a
[47]	83.0	74.3	n/a
Ours_{avg}	85.7	77.1	65.1
Ours_{max}	83.9	75.8	63.9
Ours_{att}	**87.2**	**78.4**	**65.7**

TABLE VII
INDIVIDUAL CLASS ACCURACY (%) PERFORMANCE ON STANFORD-BACKGROUND

Class	Sky	Tree	Road	Grass	Water	Building	Mountain	Foreground
[44]	92.6	78.7	92.0	84.5	62.0	80.1	13.4	67.0
[42]	**95.0**	76.0	92.0	**87.0**	68.0	85.0	29.0	63.0
[47]	89.0	80.0	90.0	82.0	66.0	83.0	**31.0**	74.0
Ours_{avg}	91.7	79.8	93.4	82.8	84.3	88.1	13.2	83.3
Ours_{max}	90.1	79.3	**94.5**	82.3	82.2	86.9	11.2	79.7
Ours_{att}	92.4	**81.1**	94.3	85.2	**89.6**	**90.3**	11.0	**83.6**

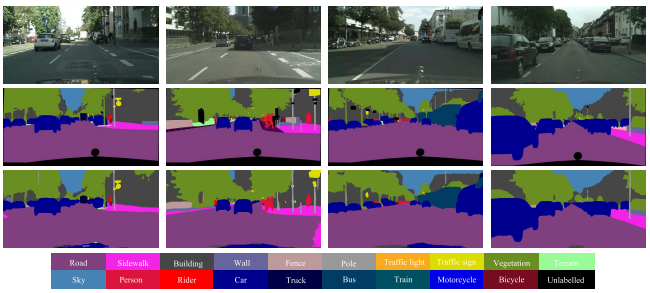

Fig. 6. Qualitative labeling results on Cityscapes. **First row:** input images. **Second row:** ground truth. **Third row:** our prediction labels.

H. Discussion on Computation Time

To further analyze the performance of the proposed approach, we compare the computation time at test phase under different conditions. In our experiments, the input images are resized to 384×384. Due to the limited GPU memory, for KITTI [21] and Cityscapes [34] datasets, we segment the input images into three image patches, and then resize them to 384×384. In evaluation, the outputs are resized to the size of input images. For KITTI [21] and Cityscapes [34] datasets, the final outputs are obtained by the combination of input image patches. The computation time is demonstrated in Table X.

From Table X, we can see that using more complex context features increases computation time at test phase, and the extraction of topic feature is slight slower than that of global feature. Compared to our method with simple max-pooling (0.48 seconds) and average-pooling (0.45 seconds), the approach with the attention model cost more computation time (0.70 seconds) because the attention model consists of two extra convolutional layers.

I. Discussion on Limitations

In this paper, we propose the ML-RNNs to model the dependencies among image units for scene labeling. Though the ML-RNNs can improve the performance, there exist two situations where misclassifications of pixels still happen. In specific, one case is that similar objects of different categories are close to each other and even overlapped. As shown in the Figure 7(a), the pole (shown in the red rectangle) is very similar to the building and overlapped with it. In this case, the proposed method misclassifies the pole into the building class because their appearances are too similar to be discriminated by the context information. Another case is that the objects are too small. In the blue rectangles in Figure 7(a), the small traffic signs lose much information in networks due to pooling operations. Without enough information, it is difficult to use context to parse them out.

TABLE VIII
QUANTITATIVE RESULTS (%) AND COMPARISONS ON CITYSCAPES

Method	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic light	Traffic sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	Mean IoU
[11]	**97.9**	81.3	90.4	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	**59.8**	93.7	**56.5**	**67.5**	**57.5**	**57.7**	**68.8**	70.4
[41]	**97.9**	**82.1**	90.7	45.2	50.4	**58.9**	62.6	68.4	91.9	69.4	94.2	78.5	**59.8**	93.4	52.3	60.8	53.7	49.9	64.2	69.7
[56]	97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8	69.4	93.7	78.9	55.0	93.3	45.5	53.4	47.7	52.2	66.0	67.1
[33]	97.5	78.5	89.5	40.4	45.9	51.1	56.8	65.3	91.5	69.4	94.5	77.5	54.2	92.5	44.5	53.4	49.9	52.1	64.8	66.8
[29]	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
Ours$_{att}$	97.8	81.0	**91.0**	**50.3**	**52.4**	56.7	**65.7**	**71.4**	**92.2**	**69.6**	**94.6**	**80.2**	59.3	**93.9**	51.1	67.6	54.5	55.1	68.6	**71.2**

TABLE IX
COMPARISONS OF DIFFERENT CONTEXT FEATURES ON FOUR DATASETS

Dataset	Context Feature	Pixel Accuracy	Class Accuracy
Camvid [3]	local (baseline)	87.60%	74.60%
	local + global	90.4%	75.4%
	local + topic	89.9%	75.8%
	local + global + topic	**91.9**%	**77.2**%
KITTI [21]	local (baseline)	88.6%	66.4%
	local + global	91.8%	68.3%
	local + topic	90.7%	68.5%
	local + global + topic	**92.1**%	**70.9**%
SiftFlow [27]	local (baseline)	83.9%	54.3%
	local + global	85.4%	54.8%
	local + topic	86.3%	56.3%
	local + global + topic	**86.9**%	**57.7**%
Stanford -background [24]	local (baseline)	85.7%	76.2%
	local + global	86.4%	77.5%
	local + topic	86.7%	77.7%
	local + global + topic	**87.2**%	**78.4**%

TABLE X
ANALYSIS OF COMPUTATION TIME (SECOND) AT TEST PHASE

Method	Context Feature	Computation time
Ours$_{avg}$	local	0.17 s
	local + global	0.24 s
	local + topic	0.35 s
	local + global + topic	0.45 s
Ours$_{max}$	local	0.15 s
	local + global	0.25 s
	local + topic	0.35 s
	local + global + topic	0.48 s
Ours$_{att}$	local	0.38 s
	local + global	0.46 s
	local + topic	0.54 s
	local + global + topic	0.70 s

(a) (b) (c)

Fig. 7. Demonstration of misclassification situations. The red rectangle contains a pole and the blue rectangles contain very small traffic signs. (a) Input image. (b) Groundtruth. (c) Prediction result.

V. CONCLUSION

In this paper, we propose the ML-CRNNs for scene labeling by exploiting dependencies among image units in different levels. We first introduce our CRNNs which are capable of capturing both long-range local, global and topic contexts in an image. To exploit different dependence relationships at multiple levels (e.g., lower levels contain more spatial dependencies while higher levels consist of more semantic dependencies), we insert our CRNNs into CNNs to model both spatial and semantic dependencies among image units. In addition, we use an attention model to learn how much attention to pay to different levels and propose our ML-CRNNs. Experiments on five benchmarks evidence the effectiveness of our approach.

REFERENCES

[1] J. M. Álvarez, M. Salzmann, and N. Barnes, "Exploiting large image sets for road scene parsing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2456–2465, Sep. 2016.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015, pp. 1–15.

[3] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.

[4] S. Bell, C. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. CVPR*, 2016, pp. 2874–2883.

[5] S. R. Bulò and P. Kontschieder, "Neural decision forests for semantic image labelling," in *Proc. CVPR*, 2014, pp. 81–88.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[7] C. Cadena and J. Košecká, "Semantic segmentation with heterogeneous sensor coverages," in *Proc. ICRA*, 2014, pp. 2639–2645.

[8] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with LSTM recurrent neural networks," in *Proc. CVPR*, 2015, pp. 3547–3555.

[9] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. CVPR*, 2012, pp. 3642–3649.

[10] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. CVPR*, 2016, pp. 3640–3649.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7913730/

[12] L. Cao, J. Luo, F. Liang, and T. S. Huang, "Heterogeneous feature machines for visual recognition," in *Proc. ICCV*, 2009, pp. 1095–1102.

[13] J. Dai, K. He, and J. Sun, "BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proc. ICCV*, 2015, pp. 1635–1643.

[14] H. Fan and H. Ling, "SANet: Structure-aware network for visual tracking," in *Proc. CVPR Workshop*, 2017, pp. 42–49.

[15] H. Fan, X. Mei, D. Prokhorov, and H. Ling, "RGB-D scene labeling with multimodal recurrent neural networks," in *Proc. CVPR Workshop*, 2017, pp. 9–17.

[16] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[17] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[18] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. ICML Workshop*, 2012.

[19] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Proc. NIPS*, 2009, pp. 545–552.
[20] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.
[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. CVPR*, 2012, pp. 3354–3361.
[22] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proc. ECCV*, 2016, pp. 519–534.
[23] J. Gao, Q. Wang, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection," in *Proc. ICRA*, 2017, pp. 219–224.
[24] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. ICCV*, 2009, pp. 1–8.
[25] S. Gould, J. Zhao, X. He, and Y. Zhang, "Superpixel graph label transfer with learned distance metric," in *Proc. ECCV*, 2014, pp. 632–647.
[26] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. NIPS*, 2011, pp. 109–117.
[27] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *Proc. CVPR*, 2009, pp. 1972–1979.
[28] G. Lin, C. Shen, A. Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. CVPR*, 2016, pp. 3194–3203.
[29] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.
[30] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr, "What, where and how many? Combining object detectors and CRFs," in *Proc. ECCV*, 2010, pp. 424–437.
[31] M. Liang, X. Hu, and B. Zhang, "Convolutional neural networks with intra-layer recurrent connections for scene labeling," in *Proc. NIPS*, 2015, pp. 937–945.
[32] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
[33] Z. Liu, X. Li, P. Luo, C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proc. ICCV*, 2015, pp. 1377–1385.
[34] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, 2016, pp. 3213–3223.
[35] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. ICCV*, 2015, pp. 1520–1528.
[36] M. Najafi, S. T. Namin, M. Salzmann, and L. Petersson, "Sample and filter: Nonparametric Scene parsing via efficient filtering," in *Proc. CVPR*, 2016, pp. 607–615.
[37] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. ICML*, 2016, pp. 1747–1756.
[38] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[39] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection," in *Proc. CVPR*, 2015, pp. 390–399.
[40] P. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. ICML*, 2014, pp. 82–90.
[41] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "Efficient convNet for real-time semantic segmentation," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2017, pp. 1789–1794.
[42] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *Proc. CVPR*, 2012, pp. 2759–2766.
[43] A. Sharma, O. Tuzel, and M.-Y. Liu, "Recursive context propagation network for semantic scene labeling," in *Proc. NIPS*, 2014, pp. 2447–2455.
[44] B. Shuai, G. Wang, Z. Zuo, B. Wang, and L. Zhao, "Integrating parametric and non-parametric models for scene labeling," in *Proc. CVPR*, 2015, pp. 4249–4258.
[45] B. Shuai, Z. Zuo, B. Wang, and G. Wang, "DAG-recurrent neural networks for scene labeling," in *Proc. CVPR*, 2016, pp. 3620–3629.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.
[47] M. Seyedhosseini and T. Tasdizen, "Semantic image segmentation with contextual hierarchical models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 951–964, May 2016.
[48] N. Souly and M. Shah, "Scene labeling using sparse precision matrix," in *Proc. CVPR*, 2016, pp. 3650–3658.
[49] P. Sturgess, K. Alahari, L. Ladický, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proc. BMVC*, 2009, pp. 1–11.
[50] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *Proc. ECCV*, 2010, pp. 352–365.
[51] J. Tighe and S. Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors," in *Proc. CVPR*, 2013, pp. 3001–3008.
[52] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. ACM MM*, 2015, pp. 689–692.
[53] F. Visin *et al.*, "ReSeg: A recurrent neural network-based model for semantic segmentation," in *Proc. CVPR Workshop*, 2016, pp. 41–48.
[54] G. Wang, D. Hoiem, and D. Forsyth, "Building text features for object image classification," in *Proc. CVPR*, 2009, pp. 1367–1374.
[55] Q. Wang, J. Gao, and Y. Yuan, "A joint convolutional neural networks and context transfer for street scenes labeling," *IEEE Trans. Intell. Transp. Syst.* [Online]. Available: http://ieeexplore.ieee.org/document/8012463/
[56] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016, pp. 1–13.
[57] J. Yang, B. Price, S. Cohen, and M.-H. Yang, "Context driven scene parsing with attention to rare classes," in *Proc. CVPR*, 2014, pp. 3294–3301.
[58] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *Proc. ECCV*, 2010, pp. 708–721.
[59] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor fusion for semantic segmentation of urban scenes," in *Proc. ICRA*, 2015, pp. 1850–1857.
[60] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. ICCV*, 2015, pp. 1529–1537.
[61] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific fusion for image retrieval," in *Proc. ECCV*, 2012, pp. 660–673.
[62] Z. Zuo *et al.*, "Learning contextual dependence with convolutional hierarchical recurrent neural networks," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 2983–2996, Jul. 2016.

Heng Fan received the B.E. degree from the College of Science, Huazhong Agricultural University, Wuhan, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, Temple University, Philadelphia, PA, USA. His research interests include computer vision, pattern recognition, and machine learning.

Xue Mei received the B.S. degree in electrical engineering from University of Science and Technology of China, Hefei, China, and the Ph.D. degree in electrical engineering from University of Maryland at College Park, College Park, MD, USA. From 2008 to 2012, he was with the Automation Path-Finding Group, Assembly and Test Technology Development and Visual Computing Group, Intel Corporation, Santa Clara, CA, USA. He is currently a Senior Research Scientist with the Department of Future Mobility Research, Toyota Research Institute, Ann Arbor, MI, USA—a Toyota Technical Center Division. His current research interests include computer vision, machine learning, and robotics with a focus on intelligent vehicles research.

Danil Prokhorov was a Research Engineer with the St. Petersburg Institute for Informatics and Automation, Russian Academy of Sciences, Saint Petersburg, Russia. He has been involved in automotive research since 1995. He was an Intern with the Scientific Research Laboratory, Ford Motor Company, Dearborn, MI, USA, in 1995. In 1997, he became a Research Staff Member with Ford Motor Company, where he was involved in application-driven research on neural networks and other methods. Since 2005, he has been with the Toyota Technical Center, Ann Arbor, MI, USA. He is currently in charge of the Department of Future Mobility Research, Toyota Research Institute, Ann Arbor.

Haibin Ling received the B.S. and M.S. degrees from Peking University, in 1997 and 2000, respectively, and the Ph.D. degree from University of Maryland, in 2006. From 2000 to 2001, he was an Assistant Researcher with Microsoft Research Asia. From 2006 to 2007, he held a post-doctoral position with the University of California at Los Angeles. He joined Siemens Corporate Research as a Research Scientist. Since 2008, he has been with Temple University, where he is currently an Associate Professor. He was a recipient of the Best Student Paper Award at ACM UIST 2003 and NSF CAREER Award in 2014. He serves as an Associate Editor for IEEE TRANSACTIONS ON PAMI and *Pattern Recognition*. He has served as the Area Chairs for CVPR 2014 and CVPR 2016.